

Perancangan Sistem Repositori Tugas Akhir Menggunakan Progressive Web App (PWA)

Engineering of Thesis Repository System Using Progressive Web App (PWA)

Aminudin^{1*}, Basri Basren², Ilyas Nuryasin³,

^{1,2,3}Jurusan Teknik Informatika, Universitas Muhammadiyah Malang, Malang
e-mail: ¹aminudin2008@umm.ac.id, ²basri.basren@gmail.com, ³ilyas@umm.ac.id

Abstrak

Penggunaan perangkat lunak seperti DSpace, EPrints, ataupun Senayan Library Information Management Systems (SLIMS) dalam membangun sistem repositori tugas akhir menghasilkan sistem yang tidak memiliki karakteristik web modern. Karakteristik yang tidak dimiliki di antaranya tidak bergantung pada konektivitas jaringan sehingga dapat diakses secara offline, terlihat seperti aplikasi native pada umumnya, dapat diinstal pada homescreen perangkat pengguna, dan menampilkan splashscreen ketika awal aplikasi dibuka. Oleh karena itu, untuk membangun sistem repositori tugas akhir yang memiliki karakteristik web modern maka sistem ini akan dibangun berbasis Progressive Web App (PWA) dengan menggunakan teknologi service worker, manifest web app, dan architecture app shell. Dari Hasil implementasi PWA pada sistem menunjukkan bahwa dengan menggunakan service worker akan membuat website tidak bergantung pada konektivitas jaringan sehingga dapat diakses secara offline. Sedangkan, dengan menggunakan manifest web app, website dapat memunculkan pop-up add to homescreen (A2HS) yang berguna bagi pengguna untuk melakukan instalasi sistem pada homescreen perangkat dan memunculkan splashscreen ketika pengguna mengakses sistem melalui icon pada homescreen perangkat. Selain itu, dengan mengimplementasikan architecture app shell membuat perpindahan halaman pada website tidak melakukan load ulang halaman secara keseluruhan. Dari hasil pada pengujian kualitas PWA berdasarkan baseline progressive web app checklist dihasilkan skor 92.4/99. Skor ini menunjukkan bahwa PWA telah diimplementasikan dengan baik pada sistem repositori tugas akhir yang dibangun.

Kata kunci—Progressive Web App (WAP), sistem repositori, service worker

Abstract

The use of software such as DSpace, EPrints, or Senayan Library Information Management Systems (SLIMS) in developing a thesis repository system, produce a system that doesn't have the characteristics of modern web including not being dependent on network connectivity so that it can be accessed offline, looks like native application in general, can be installed on home screen device and can also displays the splash screen when the application start. Therefore, to develop a thesis repository system that has modern web characteristics, this system will be developed based on a Progressive Web App (PWA) using service worker technology, web manifest app, and shell app architecture. From the results of the PWA implementation on the system shows that using a service worker will make the website not dependent on network connectivity so that it can be accessed offline. In addition, using the manifest web app on website can bring up a pop-up add to home screen (A2HS) that is useful for users to install the system on the device's home screen and bring up the splash screen when the user accesses the system via the icon on the device's home screen. And finally, by implementing the shell app architecture, making the pages on the website not reload the page as a whole. From the results on the PWA quality testing based on the progressive web app checklist produced score 92.4 / 99. This score shows that the PWA has been implemented properly in the thesis repository system that was developed.

Keywords—Progressive Web App (WAP), repository system, service worker

1. PENDAHULUAN

Perkembangan teknologi akhir-akhir ini mengakibatkan banyaknya kemudahan yang dapat dirasakan. Salah satu diantaranya adalah mudahnya dalam pembuatan serta penyimpanan dokumen digital baik berformat teks, audio, ataupun video. Salah satu media yang dapat digunakan untuk melakukan penyimpan dokumen digital adalah repositori digital. Repositori digital merupakan sebuah media yang disediakan untuk penyimpanan arsip baik yang bersifat tetap atau sementara [1]. Institutional repository yang merupakan repositori pada perguruan tinggi adalah konsep dalam mengumpulkan, mengelola, serta menyalurkan dan melestarikan karya ilmiah yang dihasilkan dari civitas akademik pada suatu perguruan tinggi. Karya ilmiah yang dimaksud disini umumnya dapat berupa artikel dari jurnal penelitian baik sebelum maupun setelah dicetak, digital format dari sebuah tugas akhir, tesis atau disertasi. Karya ilmiah tersebut akan dikelola dan disalurkan dalam bentuk digital sehingga dapat dimanfaatkan kembali untuk menunjang kegiatan akademik [2][3].

Membangun institutional repository dapat dilakukan dengan beberapa cara salah satunya dengan menggunakan sumber daya atau perangkat lunak gratis yang sudah ada seperti EPrints, DSpace, ataupun SLIMS (*Senayan Library Information Management Systems*). Selain menggunakan perangkat lunak gratis yang sudah ada tersebut, institutional repository dapat dibangun dari awal dengan arti bahwa sistem yang dibutuhkan dibuat sendiri dari awal [2]. Dari 69 institusi di Indonesia yang terdaftar pada OpenDOAR sekitar 53 institusi menggunakan Eprints untuk pengelolaan karya ilmiah, sementara pengguna 9 institusi perguruan tinggi menggunakan Dspace dan 7 institusi perguruan tinggi menggunakan perangkat lunak lainnya [4].

Universitas Muhammadiyah Malang masuk ke dalam daftar OpenDOAR yang menggunakan Eprints untuk pengelolaan karya ilmiahnya. Kelemahan yang terdapat pada pembangunan institutional repository menggunakan sumber daya seperti Eprint, Dspace dan lainnya adalah tidak didukungnya atau tidak memiliki keunggulan web modern seperti dapat diakses secara offline, terlihat seperti aplikasi native dimana setiap perpindahan halaman pada aplikasi tidak mereload halaman secara keseluruhan, dapat diinstal pada homescreen perangkat pengguna, dan menampilkan splashscreen ketika awal aplikasi dibuka melalui icon pada homescreen perangkat pengguna [5]. Oleh karena itu, untuk mendukung website sehingga memiliki beberapa keunggulan web modern tersebut maka pembangunan sistem ini akan dibangun dari awal berbasis progresive web app dengan menggunakan teknologi service worker, manifest web app, dan architecture app shell.

Progressive web app (PWA) adalah sebuah ide yang pertama kali didukung oleh insinyur Google Alex Russell pada bulan Juni 2015 [6]. PWA menggabungkan hal terbaik dari web dengan fitur yang telah dimiliki oleh aplikasi native. Beberapa hal yang dapat dilakukan dengan PWA antara lain website dapat diinstal pada ponsel pengguna dan telah ditambahkan ke homescreen layaknya aplikasi native, website yang telah diinstal dapat berfungsi secara offline dengan menggunakan offline cache APIs, dan cache aplikasi akan diperbaharui ketika ada pembaharuan dari sisi server [7].

Penelitian yang dilakukan oleh Wibowo (2017) yang mengimplementasikan PWA pada pengembangan Sistem Informasi Manajemen Mutu (SIMUTU), menghasilkan mobile web application SIMUTU yang lebih ringan, lebih cepat, fitur responsive, sampai kemampuan keamanan tingkat tinggi dan mampu diakses secara offline. Sedangkan, pada penelitian yang dilakukan oleh Karpagam, dkk (2017) yang berjudul "Performance Enhancement of Webpage Using Progressive Web App Features", menghasilkan kesimpulan bahwa dengan PWA akan meningkatkan halaman web tradisional menjadi cepat, andal, dan menarik. Halaman web yang ditambahkan service worker dapat bekerja secara offline, membantu dalam fitur push notification dan membuat icon di desktop sehingga memberikan pengalaman seperti aplikasi native [6]. Pada penelitian lainnya dilakukan oleh Mishra (2016) yang mereview tentang progressive web app dengan kesimpulan bahwa PWA dapat dimuat dengan cepat bahkan ketika pengguna berada dalam koneksi jaringan internet yang buruk [8][9].

Penelitian yang dilakukan oleh Rebecca (2017) yang berjudul “Comparing Progressive Web App with Native Android Application – an evaluation of performance when it comes to response time”. Penelitian ini dilakukan dengan membandingkan PWA dan NAA dalam hal kecepatan dalam mengakses kamera dan mengambil foto. Selain itu, juga dilakukan perbandingan dalam pengaksesan lokasi pengguna dan membuat peta. Kesimpulan pada penelitian ini menyatakan bahwa NAA masih memiliki response time yang lebih baik dari PWA dalam mengakses kamera dan mengambil foto. Namun, response time PWA mengalahkan NAA pada mengakses lokasi pengguna dan membuat peta [10].

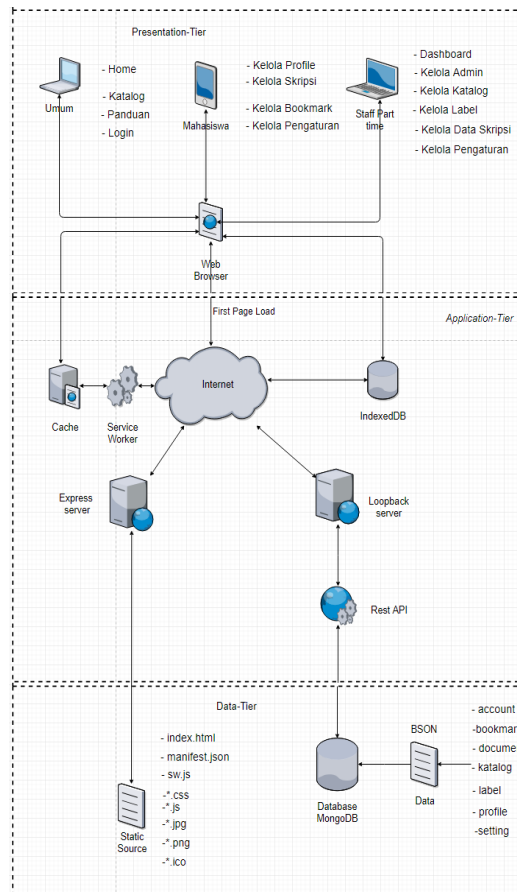
Berdasarkan masalah diatas, maka dalam penelitian ini akan dibangun web app sistem repositori tugas akhir berbasis progressive web app (PWA) dengan studi kasus yang diambil pada ruang baca Fakultas Teknik Universitas Muhammadiyah Malang. Penelitian ini diharapkan dapat menghasilkan sebuah sistem repositori tugas akhir berbasis progressive web app yang memiliki kemampuan web modern diantaranya dapat diakses secara offline, terlihat seperti aplikasi native, dapat diinstal pada homescreen perangkat pengguna dan dapat menampilkan splashscreen ketika awal aplikasi dibuka.

2. METODE PENELITIAN

2.1 Analisa Sistem

Tahap analisa sistem ini menghasilkan arsitektur sistem yang akan dibangun, dan usecase diagram yang menggambarkan kebutuhan fungsional pada sistem secara umum, serta daftar kebutuhan non-fungsional pada sistem. Pada Gambar 1 merupakan hasil arsitektur sistem yang akan dibangun menggunakan model three-tier. Arsitektur dibagi menjadi 3 bagian atau 3-tier diantaranya *presentation-tier*, *application-tier*, dan *data-tier*. Pada gambar 2 diatas merupakan hasil perancangan arsitektur pada sistem. Pada *presentation-tier* menggambarkan antarmuka pengguna yang dapat diakses oleh pengguna umum, staf dan mahasiswa melalui *web browser*. Setiap akses pengguna terhadap sistem dibedakan berdasarkan role yang dimiliki. Pengguna umum dapat mengakses halaman home, halaman katalog, halaman panduan dan halaman login. Mahasiswa dapat mengakses halaman kelola profil, halaman kelola tugas akhir, halaman kelola bookmark, halaman kelola pengaturan dan semua halaman yang dapat diakses oleh pengguna umum kecuali halaman login. Staf dapat mengakses halaman dashboard, halaman kelola admin, halaman kelola katalog, halaman kelola label, halaman kelola data tugas akhir dan halaman kelola pengaturan.

Application-tier terdiri dari *service worker* yang akan mengontrol file static yang akan disimpan di cache storage dan juga mengontrol data serta menyimpannya di indexedDB pada browser client. Selain itu, pada *application-tier* juga terdapat dua server yaitu express server dan loopback server. Express server merupakan server yang digunakan untuk menangani static resource pada client-side sistem seperti file .html, .css dan .js. Sedangkan, loopback server merupakan server yang digunakan untuk menangani REST API yang merupakan penghubung antara *presentation-tier* dan *data-tier*. Request untuk menambah, mengubah, ataupun menghapus data pada database yang terdapat pada *data-tier*, harus melalui REST API dengan mengirimkan request HTTP beserta alamat URL endpoint.



Gambar 1 Arsitektur Three Tier Sistem Repositori Tugas Akhir

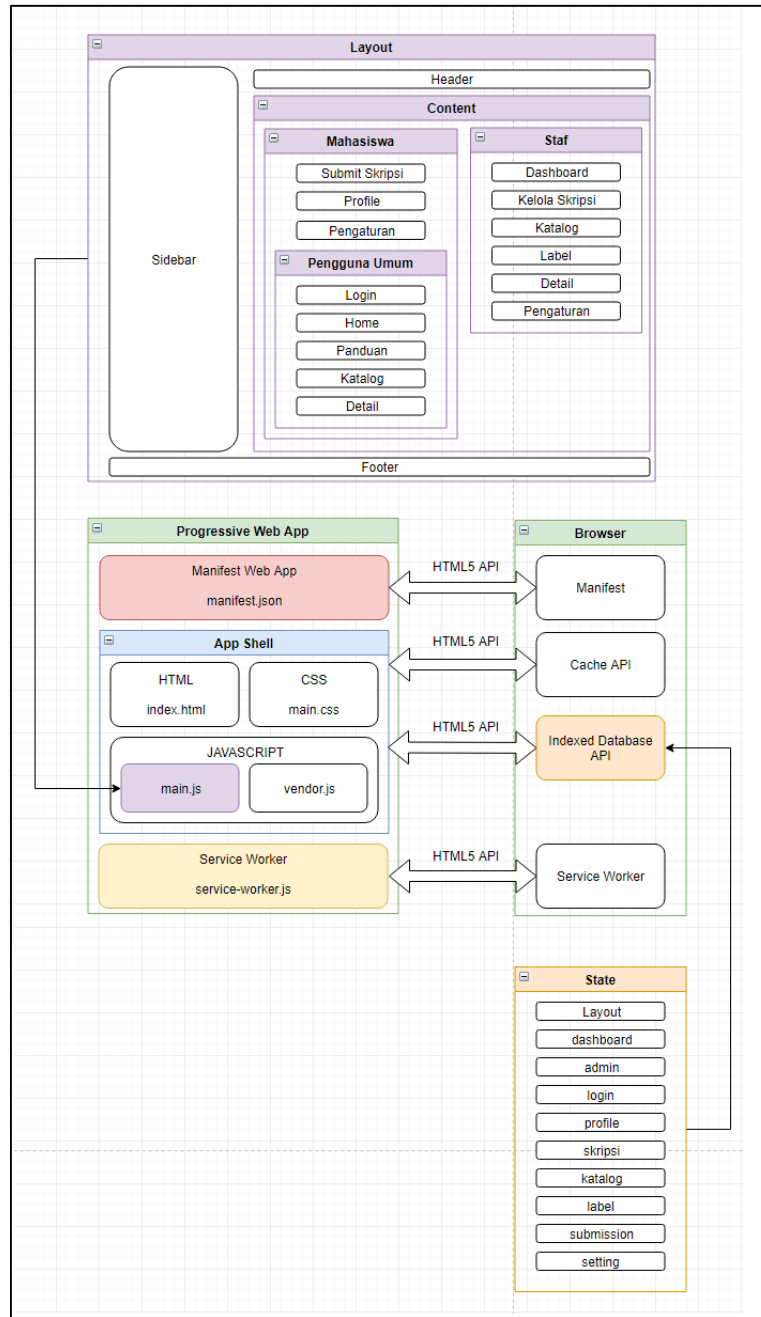
Data-tier berisi kumpulan basis data yang digunakan dalam hal ini digunakan basis data NoSQL document-oriented MongoDB. Dokumen pada MongoDB direpresentasikan dengan format JSON biner-encode yang disebut BSON. BSON memperluas model JSON dengan menyediakan tipe data tambahan, ordered field, dan efisiensi untuk encoding dan decoding dalam bahasa yang berbeda.

2.2 Rancangan Sistem

Tahap ini menghasilkan beberapa perancangan menggunakan UML diagram yang terdiri dari *activity diagram*, *sequence diagram*, dan *class diagram*. Selain itu juga dihasilkan perancangan antarmuka pada sistem, perancangan database pada sistem, dan perancangan progressive web app pada sistem yang terdiri dari perancangan architecture app shell, pemilihan strategi pada service worker, dan perancangan manifest web app.

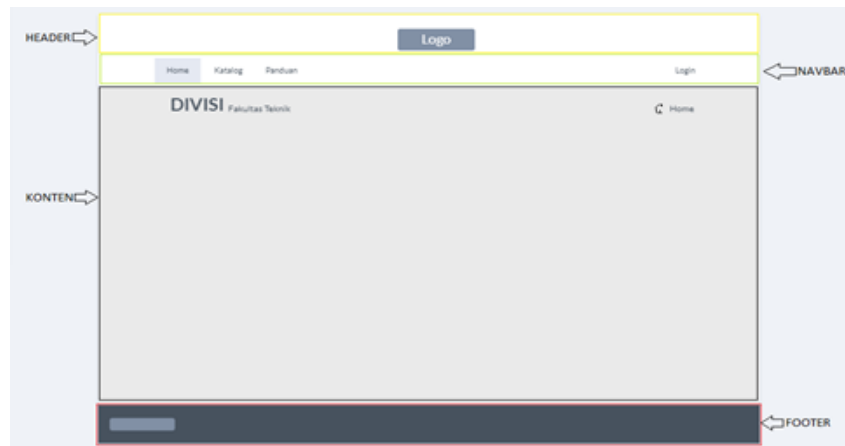
Antarmuka pada sistem akan dibangun secara responsive, diharapkan website dapat diakses pada perangkat dengan berbagai ukuran. penggunaan teknologi service worker diharapkan membuat website tidak bergantung pada konektivitas jaringan dan selalu up-to-date. Menggunakan manifest web app diharapkan akan membuat website bersifat installable, dan perancangan sistem dengan gaya architecture app shell diharapkan akan membuat website terasa seperti aplikasi native pada umumnya.

Pada Gambar 2 di bawah ini merupakan hasil perancangan arsitektur progressive web app pada sistem repositori tugas akhir pada ruang baca Fakultas Teknik UMM.



Gambar 2 Arsitektur Progressive Web App

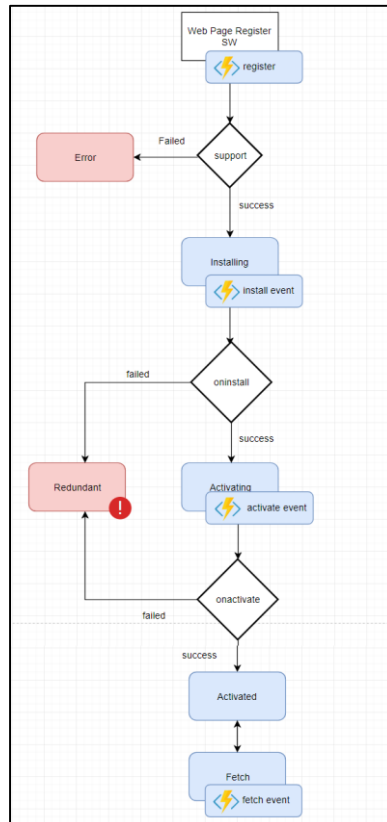
Arsitektur progressive web app pada sistem akan terdiri dari manifest web app, architecture app shell, dan service worker. Dengan menggunakan API yang disediakan HTML5, akan di daftarkan file `manifest.json` pada browser sehingga membuat website dapat memunculkan pop-up add to home screen. Pada Gambar 3 dibawah ini merupakan hasil perancangan architecture app shell pada sistem dimana akan terdapat file HTML, CSS dan Javascript yang dibutuhkan oleh website. Pada file javascript akan berisi layout dan setiap content pada website. Layout pada sistem akan terdiri dari header, sidebar, content, dan footer.



Gambar 3 Rancangan App Shell

Architecture app shell membuat web terasa seperti aplikasi di mana ketika terjadi perpindahan halaman pada website, sistem hanya mereload bagian content pada layout dan data yang dibutuhkan oleh halaman tersebut. Sedangkan, untuk bagian sidebar, header, dan footer yang terdapat pada layout bersifat tetap dan hanya di load sekali, yaitu ketika website dibuka untuk pertama kali. Pada sistem ini akan dilakukan penyimpanan static file pada cache storage sebagai media penyimpanannya dan state atau data yang dibutuhkan oleh halaman menggunakan indexeddb sebagai media penyimpanannya pada perangkat pengguna. Sehingga, membuat ketersediaan halaman dan data meskipun website diakses dalam keadaan offline.

Service worker membuat website tidak bergantung pada konektivitas dan selalu up-to-date. Dengan mendaftarkan service worker, sistem akan menyimpan asset static pada cache storage, hal ini lah yang membuat website dapat diakses dalam keadaan offline. Setiap request akan melewati service worker, ketika response yang diinginkan terdapat pada cache storage maka response akan dikembalikan dari cache storage. Namun, jika tidak terdapat pada cache storage maka service worker akan melanjutkan request ke server. Pada Gambar 4 dibawah ini merupakan hasil perancangan siklus hidup service worker pada sistem yang akan dibangun. Dimana pada siklus hidup ini terdapat beberapa tahap yaitu register, installing, activating, activated, fetch, dan redundant.



Gambar 4 Siklus Hidup Service Worker

Pada tahap register yang merupakan tahap dalam mendaftarkan service worker pada browser pengguna, pada tahap ini sistem akan mengecek didukung tidaknya penggunaan service worker pada browser yang digunakan oleh pengguna. Pada tahap installing yaitu tahap dimana akan dilakukannya instalasi *service worker* pada browser pengguna setelah tahap register berhasil. Tahap Installing dilakukan dengan melakukan pre-cache atau menyimpan file static yang dibutuhkan oleh sistem pada cache storage. Pada tahap ini digunakan strategi on install-as a dependency.

Pada tahap activating merupakan tahap dalam mengaktifkan service worker dengan cara menjalankan activate event. Pada tahap ini akan digunakan strategi on activate, jika activate event berhasil maka *service worker* akan berstatus activated. Status activated merupakan status dimana service worker telah siap digunakan dan memiliki control penuh terhadap file yang telah di cache, Fetch event merupakan event yang digunakan *service worker* dalam melakukan kontrol pada dokumen dengan menangani request yang datang dan response yang akan dikembalikan. Untuk menangani *fetch event*, strategi yang digunakan adalah cache then network, dan Status redundant merupakan status ketika installing event atau activating event gagal di eksekusi pada *service worker*.

2.3 Rancangan Pengujian

Pengujian kualitas PWA dilakukan dengan menggunakan *lighthouse tool* v4.0.0-alpha.2-3.2.1 pada website <https://www.webpagetest.org> yang secara otomatis akan memverifikasi banyak *item* di daftar *baseline progressive web app checklist*. *Lighthouse* adalah perangkat lunak yang bersifat *opensource* untuk meningkatkan kualitas aplikasi web. *Lighthouse* bisa dijalankan sebagai ekstensi Chrome atau dari baris perintah. Dengan menginputkan URL yang ingin di periksa, *lighthouse* akan menjalankan serangkaian pengujian terhadap laman tersebut, kemudian menghasilkan sebuah laporan mengenai seberapa bagus

laman itu menjalaninya. *Lighthouse* saat ini memfokuskan pada fitur *progressive web app*, misalnya *add to homescreen* dan dukungan *offline*.

Lighthouse mengembalikan skor *Progressive Web App* (PWA) antara 0 dan 100. Skor 0 adalah skor terburuk, dan 100 adalah yang terbaik. Audit PWA didasarkan pada *baseline PWA checklist* yang berisi 15 daftar persyaratan PWA. Pemberian nilai pada *Lighthouse* didasarkan pada pemeriksaan yang dilakukan secara otomatis ke 12 dari 15 daftar persyaratan PWA. Sedangkan, untuk 3 daftar yang tersisa akan di uji secara manual dan tidak masuk dalam pemberian nilai. Masing-masing dari 12 persyaratan yang diuji secara otomatis akan diberi bobot yang sama. Jadi, pada setiap audit yang benar akan meningkatkan skor sekitar ~9 point. Pada rancangan pengujian akan dilakukan audit pada ke 15 persyaratan yang ada dan memberikan bobot yang sama, termasuk pada 3 persyaratan yang diuji secara manual. Sehingga, setiap audit yang benar akan meningkatkan skor PWA sekitar 6.6 dan total nilai adalah 99 untuk nilai terbaik dan 0 untuk nilai terburuk.

3. HASIL DAN PEMBAHASAN

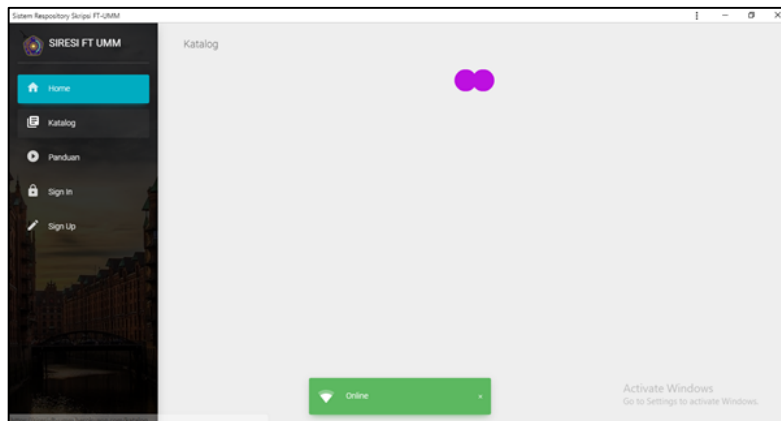
3.1 Implementasi Sistem

Pada tahap ini akan dilakukan implementasi terhadap perancangan sistem ke dalam bahasa pemrograman. Sehingga, dihasilkan sistem repositori tugas akhir berbasis PWA. Beberapa pengimplementasian sistem yang akan dibahas diantaranya pengimplementasian kebutuhan sistem yang terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dibutuhkan selama proses pembangunan sistem repositori pada ruang baca Fakultas Teknik UMM. Selain itu, akan dilakukan implementasi program yang terdiri dari implementasi framework *loopback* dalam membangun REST API pada sistem, implementasi library javascript *reactjs* dalam membangun client-side pada sistem, implementasi database menggunakan MongoDB, dan implementasi antarmuka pengguna. Juga akan dilakukan pengimplementasian PWA yang terdiri dari implementasi manifest web app, implementasi architecture app shell, dan implementasi service worker. Hasil implementasi manifest web app pada sistem repositori tugas akhir ruang baca Fakultas Teknik UMM, dapat dilihat pada Gambar 5 di bawah ini, di mana pada Gambar tersebut ditunjukkan bahwa sistem yang dihasilkan dapat menampilkan pop-up *add to homescreen* secara otomatis ketika sistem diakses melalui browser google chrome pada perangkat android.



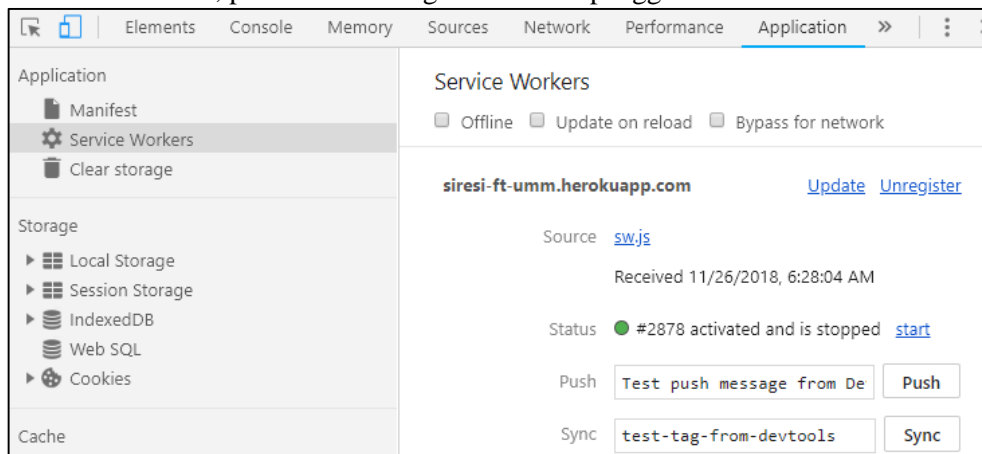
Gambar 5 Implementasi Manifest Web App

Implementasi *manifest web app* pada sistem selain dapat memunculkan pop-up add to homescreen, juga memberikan kemampuan lain pada sistem yaitu dapat memunculkan splashscreen ketika diakses melalui icon pada homescreen perangkat android. Hasil implementasi *architecture app shell* menunjukkan bahwa perpindahan pada halaman dalam website, tidak mereload halaman secara keseluruhan dan juga menampilkan indikator *loading* selama proses *load* sumber daya yang dibutuhkan pada sistem. Pada Gambar 6 di bawah ini merupakan hasil pengimplementasian *architecture app shell* pada sistem repositori tugas akhir ruang baca Fakultas Teknik UMM.



Gambar 6 Implementasi Arsitektur App Shell

Hasil implementasi service worker pada sistem yang dibangun dapat dilihat pada Gambar 7 di bawah ini. Dimana pada gambar ditunjukkan bahwa service worker telah berhasil di daftarkan pada browser pengguna. Selain itu, service worker juga berhasil menyimpan file static yang dibutuhkan sistem, pada cache storage di browser pengguna.



Gambar 7 Implementasi Service Worker

3.2 Pengujian

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun. Pengujian dilakukan terhadap kualitas PWA pada sistem berdasarkan baseline progressive web app checklist. Pada pelaksanaannya, pengujian kualitas PWA ini akan dilakukan secara manual dan menggunakan alat bantu lighthouse tool. Pada tabel 2 dibawah ini merupakan hasil pengujian kualitas progressive web app pada sistem repositori tugas akhir ruang baca Fakultas Teknik UMM.

Tabel 1 Hasil Pengujian Kualitas PWA pada Sistem Berdasarkan Baseline Progressive Web App checklist

| No. | Baseline PWA Checklist | Cara pengujian | Hasil | Point |
|-------|---|--|--|-------|
| 1 | Situs di sajikan di atas HTTPS | Gunakan <i>Lighthouse</i> untuk memverifikasi website disajikan melalui HTTPS | Passed | 6.6 |
| 2 | Redirect HTTP traffic to https | Gunakan <i>Lighthouse</i> untuk memverifikasi setiap HTTP traffic diredirect ke HTTPS. | Passed | 6.6 |
| 3 | Halaman responsif pada tablet & perangkat seluler | Gunakan <i>Lighthouse</i> untuk memverifikasi desain sistem mobile-friendly | Passed | 6.6 |
| 4 | Mempunyai tag <meta name="viewport"> dengan width atau initial-scale | Gunakan <i>Lighthouse</i> untuk memverifikasi halaman memiliki meta viewport dengan width dan initial-scale untuk optimisasi website pada mobile | Passed | 6.6 |
| 5 | <i>Service worker</i> berhasil didaftarkan | Gunakan <i>Lighthouse</i> untuk memverifikasi bahwa <i>service worker</i> dapat didaftarkan | Passed | 6.6 |
| 6 | Semua URL aplikasi dapat dimuat saat <i>offline</i> | Gunakan <i>Lighthouse</i> untuk memverifikasi start URL merespon dengan status 200 saat akses secara <i>offline</i> | Passed | 6.6 |
| 7 | Menampilkan konten ketika javascript tidak disediakan | Gunakan <i>Lighthouse</i> untuk memverifikasi website akan menampilkan konten ketika javascript dalam keadaan disabled. | Passed | 6.6 |
| 8 | Metadata disediakan untuk fitur "Tambahkan ke Layar Utama" | Gunakan <i>Lighthouse</i> untuk memverifikasi Pengguna dapat diminta untuk melakukan "Tambahkan ke layar Utama" | passed | 6.6 |
| 9 | Warna brand dan address bar cocok | Gunakan <i>Lighthouse</i> untuk memverifikasi warna brand dan address bar pada browser cocok | Passed | 6.6 |
| 10 | Short_name tidak terpotong pada <i>homescreen</i> | Gunakan <i>Lighthouse</i> untuk memverifikasi short_name pada website tidak terpotong | passed | 6.6 |
| 11 | Situs dapat menampilkan splash screen | Gunakan <i>Lighthouse</i> untuk memverifikasi konfigurasi splash screen pada situs | passed | 6.6 |
| 12 | First Load dapat dimuat dengan cepat bahkan pada 3G | Gunakan <i>Lighthouse</i> pada Nexus 5 (atau yang serupa) untuk memverifikasi waktu ke interaktif <10s untuk kunjungan pertama pada simulasi jaringan 3G | passed | 6.6 |
| 13 | Situs dapat dijalankan di berbagai browser (Cross-Browser) | Uji situs di Chrome browser secara manual. Akan digunakan chrome versi 71.0.3578.80 (Official Build) (64-bit) untuk Desktop dan versi 55.0.2883.91 untuk android Uji situs di Edge browser secara manual. Akan digunakan edge versi 42.17134.1.0 Uji situs di Firefox browser secara manual. Akan digunakan firefox versi 63.0.3 (64-bit) pada Desktop dan 63.0.2 pada android Uji situs di Safari browser secara manual. Akan digunakan safari very 12.0.2 Uji situs di Samsung internet secara manual. Akan digunakan Samsung internet versi 7.4.00.70 pada android Uji situs di UC browser secara manual. Akan digunakan UC Browser versi 7.0.185.1002 pada desktop dan versi 12.9.9.1155 pada android | passed passed passed passed passed | 6.6 |
| 14 | Transisi halaman terasa cepat dan tidak terasa seperti memblokir jaringan | Buka Aplikasi pada simulasi jaringan yang sangat lambat. Setiap kali Anda mengetuk tautan / tombol di aplikasi, laman harus segera merespons, baik dengan: <ul style="list-style-type: none"> • Transisi langsung ke layar berikutnya dan menampilkan layar placeholder loading screen sambil menunggu konten dari jaringan • Indikator loading ditampilkan saat aplikasi menunggu <i>response</i> dari jaringan. | Passed | 6.6 |
| 15 | Setiap halaman memiliki URL | Uji secara manual dan pastikan setiap halaman dapat di tautan melalui URL, dan URL itu unik untuk tujuan dapat dibagikan di media sosial dengan pengujian dengan setiap halaman dapat dibuka dan diakses langsung melalui jendela <i>browser</i> baru. | Failed | 0 |
| Total | | | 92.4 | |

Pada Tabel 1 diatas dapat ditunjukan bahwa sistem sudah memiliki kualitas PWA yang cukup baik dimana score yang dihasilkan adalah 92.4 dari nilai maksimal 99. Dari hasil pengujian kualitas PWA dapat disimpulkan bahwa website repositori tugas akhir pada ruang baca Fakultas Teknik UMM yang dibangun sudah memiliki beberapa karakteristik pada progressive web app diantaranya:

- a. Tampilan dapat menyesuaikan pada ukuran layar perangkat yang digunakan oleh pengguna dalam mengakses system.
- b. Mengembalikan response status 200 ketika diakses secara offline.
- c. Dapat memunculkan pop-up add to home screen sehingga dapat diinstal pada homescreen perangkat pengguna,
- d. Terlihat seperti aplikasi native pada umumnya dimana setiap perpindahan halaman tidak melakukan load ulang secara keseluruhan. selain itu, website yang diakses melalui icon pada homescreen memunculkan splashscreen,
- e. Dapat diakses di berbagai browser dengan versi terbaru seperti google chrome, firefox, Microsoft Edge, dan iOS safari.
- f. First-load pada website memerlukan waktu interaktif kurang dari 10 detik untuk kunjungan pertama pada simulasi jaringan 3G.
- g. Namun, pada pengujian terdapat 1 persyaratan yang bernilai failed yaitu pada hasil pengujian nomor 15. Persyaratan ini dianggap tidak dimiliki oleh sistem repositori tugas akhir yang telah dibangun karena setiap halaman detail katalog tidak memiliki url yang unik.

4. KESIMPULAN

Berdasarkan hasil pengujian kualitas PWA pada sistem menunjukkan score 92.4/99 dimana dari ke 15 persyaratan PWA terdapat 14 persyaratan yang dapat di implementasikan pada sistem repositori tugas akhir ruang baca Fakultas Teknik UMM. Berdasarkan hasil tersebut dapat disimpulkan bahwa PWA sudah diimplementasikan dengan baik kedalam sistem repositori tugas akhir pada ruang baca Fakultas Teknik UMM. Sehingga, sistem repositori tugas akhir ini memiliki karakteristik PWA diantaranya:

- a. Tampilan akan menyesuaikan pada ukuran layar pada perangkat yang digunakan oleh pengguna,
- b. Dapat diinstal pada homescreen perangkat pengguna yang membuat pengguna dapat mengakses dengan cepat melalui icon pada homescreen.
- c. Tidak bergantung pada konektivitas jaringan sehingga dapat diakses secara offline, dan
- d. Terlihat seperti aplikasi native pada umumnya dimana setiap perpindahan halaman pada website tidak mereload halaman secara keseluruhan.
- e. Dapat diakses di berbagai browser dengan versi terbaru seperti google chrome, firefox, Microsoft Edge, dan iOS safari.
- f. Pembaharuan pada sisi server akan secara otomatis memperbaharui apa yang dimiliki pada sisi client.
- g. First-load pada website memerlukan waktu interaktif kurang dari 10 detik untuk kunjungan pertama pada simulasi jaringan 3G

Penelitian ini masih dapat dikembangkan lebih lanjut pada penelitian berikutnya. Beberapa saran pengembangan pada penelitian ini diantaranya dengan membuat setiap url pada sistem memiliki alamat yang unik. Hal ini akan membuat kualitas PWA akan sempurna. Selain itu, juga perlu dilakukannya pengamanan akses terhadap state yang disimpan secara persisten di indexedDB pada browser pengguna. Untuk mengamankan state pada indexeddb dapat digunakan metode enkripsi [11]. Pengamanan pada setiap endpoint REST API juga perlu dilakukan. Untuk mengamankan setiap endpoint pada REST API dapat dilakukan dengan mengatur access control list (ACL) pada setiap schema.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Laboratorium Teknik Informatika yang telah menyediakan fasilitas untuk mengembangkan aplikasi web dengan menggunakan teknologi Progressive Web App (PWA). Tidak lupa pula Program Studi Teknik Informatika Universitas Muhammadiyah Malang yang telah menaungi penulis di dalam melakukan penelitian ini. begitupula Ruang Baca Fakultas Teknik yang telah memberikan kebutuhan terkait dengan rekayasa kebutuhan di dalam mengembangkan sistem PWA ini.

DAFTAR PUSTAKA

- [1] M. H. Bhat, "Open Access Repositories: A Review," *Libr. Philos. Pract.*, pp. 1–8, 2010.
- [2] P. Saputri, Z. Arifin, and Y. Yulianto, "Rancang Bangun Web Repository Skripsi Mahasiswa Berbasis OAI-PMH 2.0 Menggunakan Google App Engine (studi Kasus : Program Studi Ilmu Komputer Universitas Mulawarman)," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 11, no. 1, pp. 64–72, Feb. 2016.
- [3] A. Aminudin, "Implementasi Single Sign On (SSO) Untuk Mendukung Interaktivitas Aplikasi E-Commerce Menggunakan Protocol Oauth," *J. Gamma*, vol. 1, no. 10, 2014.
- [4] "OpenDOAR - Summaries - Indonesia."
- [5] A. Aminudin, *Cara Efektif Belajar Framework LARAVEL*, I. Yogyakarta: Lokomedia, 2015.
- [6] Karpagam, "Performance Enhancement of Webpage Using Progressive Web App Features," *Int. J. Innov. Res. Adv. Eng.*, vol. 4, no. 03, pp. 97–103, 2017.
- [7] S. K. A. Pratama, "Peluang dan Tantangan Web Developer – WWWID – Medium."
- [8] R. S. Mishra, "Progressive WEBAPP : Review," *Int. Res. J. Eng. Technol.*, vol. 03, no. 06, pp. 3028–3032, 2016.
- [9] G. D. Wibowo, "Perancangan Kuisisioner Evaluasi Mutu Berbasis Mobile Web Application Menggunakan PWA (Progressive Web App) (Studi Kasus: simutu.umm.ac.id)," Universitas Muhammadiyah Malang, 2017.
- [10] F. Rebecca and D. Alexandre, "Comparing Progressive Web Applications with Native Android Applications," 2017.
- [11] A. Aminudin, A. F. Helmi, and S. Arifianto, "Analisa Kombinasi Algoritma Merkle-Hellman Knapsack dan Logaritma Diskrit pada Aplikasi Chat," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, pp. 325–334, 2018.